Journal of Mechanics, 2026, **41**, 690–702
https://doi.org/10.1093/jom/ufaf051
Regular Article

Research on energy optimization for liquid-cooled server cooling systems based on swarm intelligence algorithms and LSTM

Hsuan-Li Shih[1], Chien-Ming Lee[2], Kai-Yang Tung[2] and Rongshun Chen[1,*]

[1]Department of Power Mechanical Engineering, National Tsing Hua University, Hsinchu, Taiwan
[2]Thermal Design Laboratory, Inventec Corporation, Taoyuan City, Taiwan

*Corresponding author: rchen@pme.nthu.edu.tw

ABSTRACT

With the growing development of artificial intelligence and the increasing importance of computational power, servers are required to be equipped with high-density graphics processing unit (GPU). The high-power consumption of GPU significantly increases the cooling demand, potentially reducing the operational efficiency of data centers. Therefore, how to optimize the energy efficiency of server cooling systems while ensuring the safe operation of GPU has become a crucial research topic. This study focuses on the energy optimization of fan and pump within the cooling distribution unit in a liquid-to-air server cooling system under high thermal load conditions. By introducing deep learning techniques, a temperature prediction model based on long short-term memory networks is constructed, and global optimization of control parameters is performed using particle swarm optimization. The proposed approach aims to minimize cooling system energy consumption while maintaining GPU temperatures within safe limits. Simulation and experimental results verify that the proposed method can effectively adjust fan and pump loads, reduce cooling energy consumption and improve the dynamic response and control stability of the system. This provides a feasible solution for future energy-efficient optimization of cooling systems in data centers.

KEYWORDS: liquid to air server cooling system, energy consumption optimization, particle swarm optimization (PSO), long short-term memory (LSTM)

1. INTRODUCTION

With the widespread application of artificial intelligence and high-performance computing, data center servers are facing unprecedented challenges. When computational loads are concentrated on high-power components such as graphics processing units (GPUs), traditional cooling control strategies often fail to respond promptly to fluctuating thermal loads, leading to decreased energy efficiency and compromised system stability. Therefore, achieving efficient dynamic control of the cooling system has become a critical issue in data center operations and management.

Qaiser *et al.* [1] highlight the unprecedented challenges faced by data centers. The 2024 Global Data Center Management Survey [2] reports a 6% increase in 7–9 kW rack density compared to 2023. Pilz and Heim [3] estimate that there are 10 000–30 000 data centers worldwide, with 335–1325 large centers (over 10 MW). Koot and Wijnhoven [4] predict that by 2030, the total energy consumption of data centers will reach 1287 TWh, equivalent to the energy consumption of a medium-sized city. Ahmed *et al.* [5] note that cooling systems are the second-largest energy consumer, accounting for 30–40% of total energy consumption.

Given the significant energy consumption of cooling systems, optimizing energy efficiency through power usage effectiveness (PUE) is crucial. While most cooling systems lack precise dynamic control for cooling distribution units (CDUs), there is still potential to lower PUE. This study proposes an intelligent control strategy that dynamically adjusts fan and pump parameters, using deep learning models to predict server temperature time series and reduce cooling energy consumption. This approach aims to provide new insights for liquid cooling technology, enhancing energy efficiency and flexibility and offering low-carbon solutions for data centers.

In 2021, Ding *et al.* [6] found that the impact of coolant and fan adjustments on server temperature exhibits high nonlinearity and temporal characteristics, influenced by the complex interactions of server structure, operating conditions and environmental factors, which traditional control methods struggle to address accurately. In the temperature prediction model section, Athavale *et al.* [7] in 2019 compared four methods—ANN, SVR, GPR and POD—for temperature prediction in data centers. The results showed that GPR performed best with small datasets, ANN was efficient in multi-output scenarios but lacked extrapolation ability, SVR had good adaptability but lower efficiency and POD demonstrated physical consistency in transient simulations but relied on the number of modes. In 2021, Asgari *et al.* [8] proposed a gray-box model combining

Received: 30 September 2025; **Revised:** 15 November 2025; **Accepted:** 20 November 2025

physical equations and data-driven methods to predict server inlet and CPU temperatures, demonstrating better stability and accuracy in extrapolation scenarios. Ham *et al.* [9] explored the relationship between inlet and exhaust temperatures, finding that increased power consumption linearly raised exhaust temperatures and increased fan energy consumption. Gupta *et al.* [10] used mechanical impedance networks and energy conservation equations to simulate internal airflow in servers, achieving prediction errors of only 3–10%. Tsai *et al.* [11] proposed a hybrid model combining data-driven and physical modeling to predict airflow and temperature distribution, achieving high accuracy and efficiency under various loads and cooling configurations. Ayala *et al.* [12] used grammar evolution techniques to build a server temperature prediction model, achieving root mean square error (RMSE) below 2.5°C. In the time series prediction model section, Verma *et al.* [13] compared autoregressive integrated moving average model (ARIMA), long short-term memory (LSTM) and Facebook Prophet models, finding that LSTM performed best in accuracy, ARIMA suited stationary data and Prophet had advantages in handling periodic and holiday effects. Sonata and Heryadi [14] analyzed the performance of LSTM and Transformer in long-term time series prediction, showing that LSTM outperformed Transformer with lower mean absolute error (MAE) and RMSE, making it more suitable for capturing long-term trends, while Transformer may be more advantageous for short-term data. In the review of the application of swarm intelligence algorithms in energy savings, Dai [15] and Jiang [16] in 2022 and 2021 applied grey wolf optimizer and whale optimization algorithm to vehicle energy management, achieving over a 30% reduction in energy consumption during the new european driving cycle (NEDC) driving cycle and improving driving range by 6–9% for hybrid vehicles while significantly reducing fuel consumption by about 19%. In 2022, Gad *et al.* [17] summarized over 2000 studies, highlighting the contributions of particle swarm optimization (PSO) in energy-saving applications across smart cities, industrial processes and environmental management, showing PSO's robustness and efficiency in high-dimensional, nonlinear and multi-objective optimization problems, further improving energy management and efficiency.

To solve the issue about CDU control, this study proposes an intelligent control strategy combining PSO and LSTM models. PSO is used to dynamically optimize pump and fan control parameters, while the LSTM model predicts the server temperature time series, enabling real-time response and energy efficiency optimization of the cooling system. This approach enhances cooling efficiency, reduces energy consumption and provides a new practical solution for high-efficiency, low-energy liquid cooling technology.

2. SYSTEM AND CONTROL METHOD

This chapter will introduce the system and control conditions used and the design of the controller.

2.1 System architecture

The server used in this study is a simulated device, equipped with copper dummy heaters to replicate the thermal effects of GPUs under high computational load; in this study, it is referred to as the server. It contains 8 dummy heaters with a maximum power of 1 kW to simulate GPUs and 6 dummy heaters with a maximum power of 600 W to simulate NV Switches, resulting in a total heat output of approximately 12 kW. However, due to water circuit limitations, the maximum usable heat is 10 kW. The 8 GPU dummy heaters are paired to form GPU1 to GPU4, and the architectural diagram is shown in Fig. 1. This study employs an external liquid-to-air cooling system in Fig. 2, where the coolant exchanges heat between the server and the CDU, and the CDU dissipates heat to the environment via fans; the CDU schematic is shown in Fig. 3. The objective of this study is to maintain GPU temperatures within a safe limit of 70°C while minimizing the total energy consumption of fans and pumps in the CDU, thereby achieving energy-efficient optimization.

To simulate varying thermal loads under different data center operating scenarios, three representative load conditions were designed: high, medium and low. Low load is set to 25% of maximum power consumption rating to represent the quiescent power consumption to maintain the GPU server function. Due to the safety issue and the controllability of the application system high load is set to 75% rather than 100% to emulate the load, while the GPU server is handling large number of requests. Medium load is set to 50% as an intermediate state between high load and low load. The heater settings for each condition are presented in Table 1, and the heat distribution is illustrated in Fig. 1. GPU2 and GPU4 are located upstream of GPU1 and GPU3. Since the four GPU dummy heaters are controlled synchronously, the temperatures of GPU1 and GPU3 are generally higher than those of GPU2 and GPU4 as the cooling water flows through. The NV Switches consume relatively low power, generating less heat than the GPU dummy heaters. In this study, particular attention is given to the temperature control of GPU1 Heater1.

2.2 Controller design—LP controller

To achieve intelligent control of dynamic cooling parameters in liquid-cooled server systems, this study designed a control architecture that combines a predictive model with an optimization algorithm. This architecture, called the LSTM-PSO controller (LP controller), is shown in Fig. 4.

Besides the PSO method, metaheuristics cover numerous algorithms such as genetic algorithm or simulated annealing algorithm. Comparison of those algorithms with PSO may be an issue, as in such a complex application system in this study, finding an objective and unique comparison standard is unfeasible. Thus, this study will focus on the optimization of the PSO method.

Although fuzzy control was considered, its rule-based nature requires extensive expert tuning and scales poorly with multiple interacting variables. Given the system's strong nonlinearity and time-delay characteristics, it may cause rule explosion and inconsistent responses. Moreover, lacking a true optimization mechanism, fuzzy control cannot ensure global energy efficiency; thus, it was not adopted.

The primary task of the controller is to iteratively select the optimal fan speed and pump flow rate settings in each control cycle using the PSO algorithm, so that the temperature of GPU1

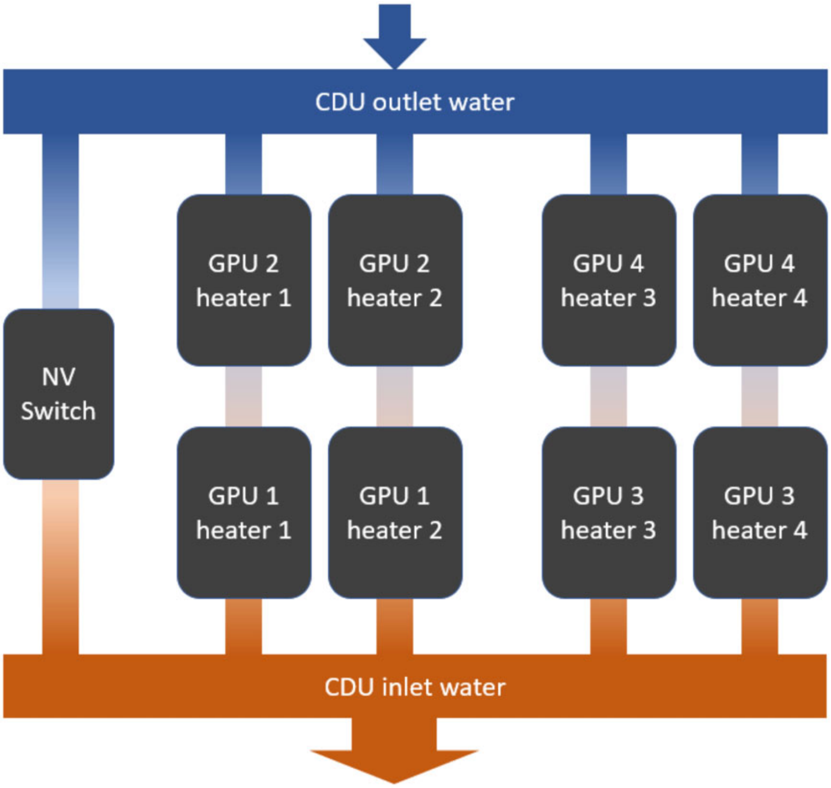


Figure 1 Server cooling circuit diagram.

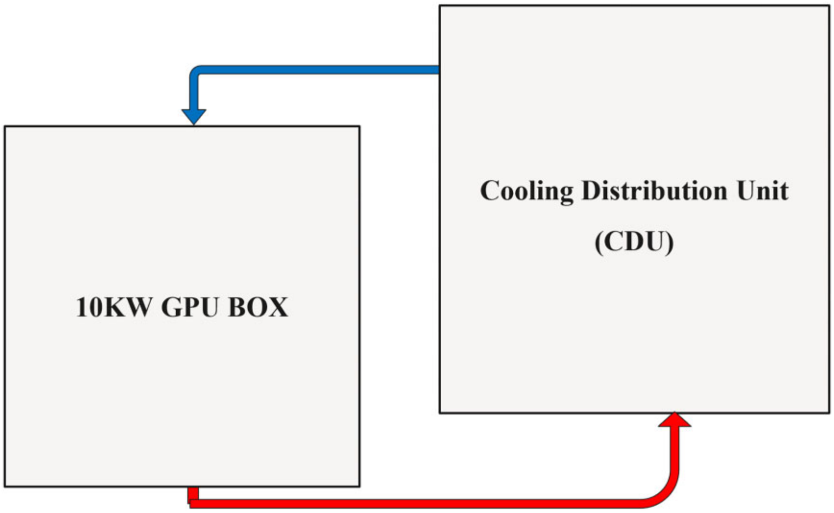


Figure 2 Schematic diagram of external circulation liquid-to-air cooling.

Heater1 is maintained stably near the safe target value while minimizing the total energy consumption of the fans and pumps in the CDU system.

As shown in the control flow in Fig. 4, the process begins with "particle initialization," where the controller randomly generates multiple candidate control parameter sets based on the current system state to serve as the initial solutions for the particle swarm. Each particle then uses the LSTM model to predict the temperature of GPU1 Heater1 at the next time step $(T+1)$ under its corresponding parameter set, and calculates the associated fan and pump power consumption based on the error between the predicted and actual temperatures. The cost function considers both temperature error and energy consumption. The optimization module follows the PSO framework, updating particle states in each generation according to the evaluation results and identifying the best-performing particle within the generation as a reference. Through iterative search and updates across multiple generations, the controller can seek a global minimum cost solution within a wide control parameter space. After the search is completed, the controller outputs the current global

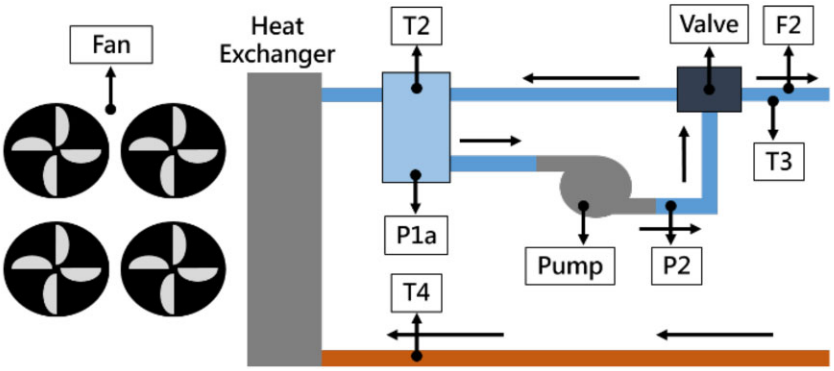


Figure 3 Schematic diagram of the interior of the coolant distribution device.

Table 1 Server heating element power setting.

Load condition	GPU power setting (each)	NV switch total power	System total power (W)
Low load	250 W (25%)	0 kW	2.5 kW (25%)
Medium load	600 W (60%)	≈0.2 kW	5.0 kW (50%)
High load	850 W (85%)	≈0.7 kW	7.5 kW (75%)

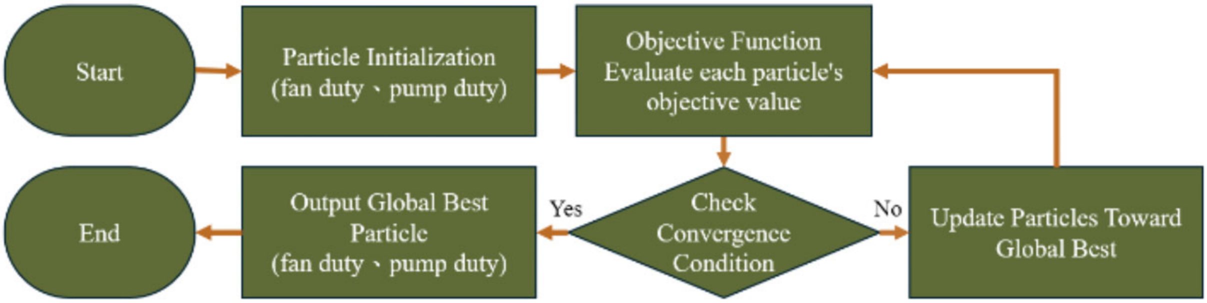

Start
Particle Initialization (fan duty、pump duty)
Objective Function Evaluate each particle's objective value
Check Convergence Condition
Yes
No
Update Particles Toward Global Best
Output Global Best Particle (fan duty、pump duty)
End

Figure 4 Controller flow chart.

optimal control parameters (i.e. fan duty and pump duty) to the physical CDU system to execute the corresponding speed settings. To account for the thermal inertia and response delay of the cooling system, the controller is set with a 10-second sampling period; after each control decision is executed, the system pauses updates and waits for the physical system response. After 10 s, it reads sensor data again, updates the internal state and initiates the next control cycle.

2.3 Objective function design

The objective function used is defined as follows:

$$f\,(p) = W_T\ \cdot |T_{model} - \ T_{target}| + W_E \cdot E_{total} + Penalty, \tag{2.1}$$

where W_T , T_{model}, T_{target}, W_E and E_{total}, represent temperature error weight, LSTM model predicts the GPU temperature at the next moment, target GPU temperature, energy consumption weight, total fan and pump power. The penalty term is used to suppress conflicting changes in fan and pump parameters. It is calculated as follows:

$$Penalty = \begin{cases} P * |\Delta_{fan}| * |\Delta_{pump}| & \text{if } \Delta_{fan} * \Delta_{pump} < 0 \\ 0 & \text{if } \Delta_{fan} * \Delta_{pump} \geq 0, \end{cases} \tag{2.2}$$

where Δ_{fan}, Δ_{pump} and P, represent fan duty variation between two adjacent control cycles, pump duty variation between two adjacent control cycles and punishment intensity coefficient.

2.4 LSTM model

The model used in this study is a two-layer LSTM layer. During training, the LSTM layer undergoes dropout before being flattened into the fully connected layer. Since only the temperature of GPU1 Heater1 at the next sampling time needs to be predicted, the output size is 1, the optimizer is Adam, and the initial learning rate is 0.01. Figure 5 shows the internal architecture of the LSTM layer.

The collection of training and validation data is shown in Table 2.

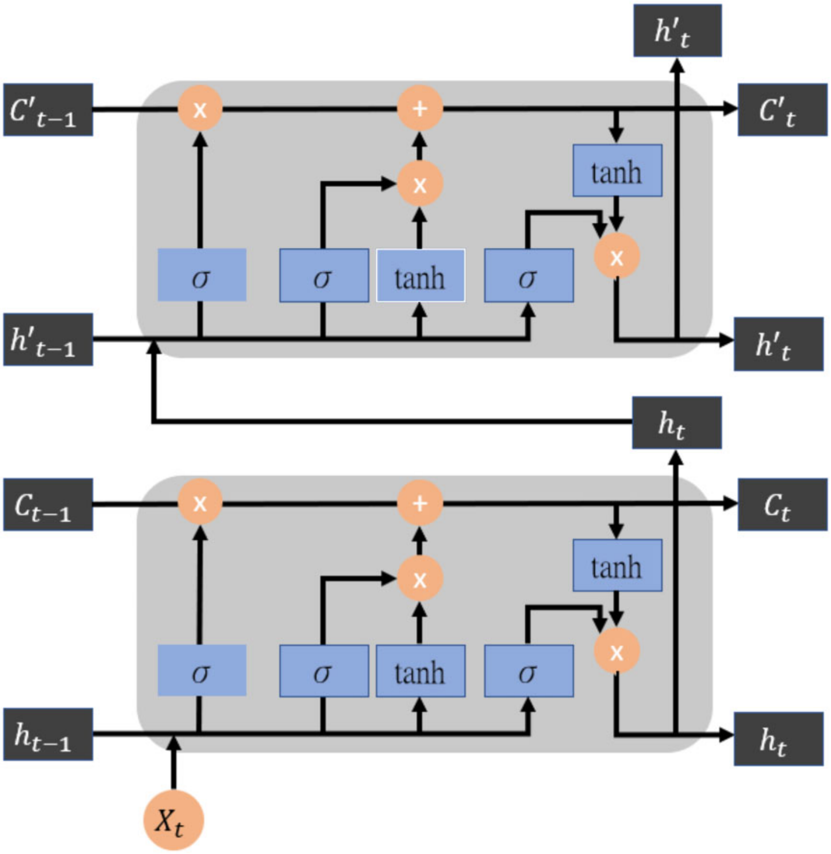


Figure 5 LSTM architecture diagram.

Table 2 Training data collection ranges.

Duty range	Fan duty	Pump duty	Corresponding load condition
Low duty	15–40%	30–60%	Medium load
Medium duty	30–60%	50–80%	Medium load, high load
High duty	50–80%	70–90%	High load

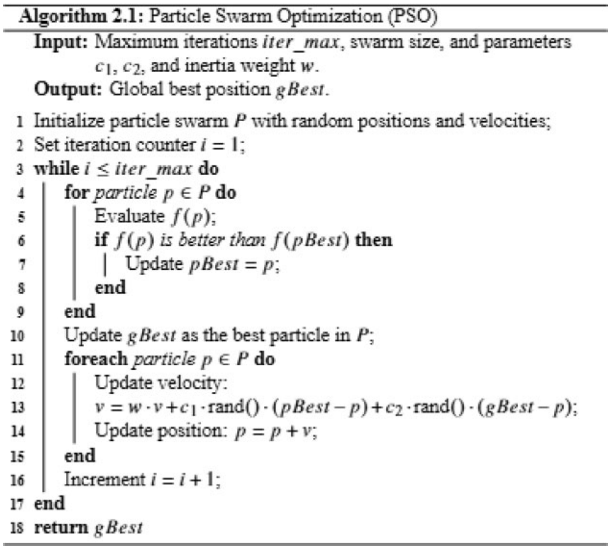
The server's dummy heaters were divided into three load zones, with the training data primarily coming from the medium- and high-load zones' temperature responses. During data collection, according to empirical rules based on the system's response range, Fan Duty and Pump Duty were randomly adjusted every 20 s within the medium- and high-load zones. These adjustments were further divided into three Duty ranges—low, medium and high—based on the load zone. The validation data were collected using the same procedure as the training data, except that adjustments were made every 30 s and the total collection duration was reduced by approximately two-thirds.

In this study, seven features were ultimately selected: GPU1 Power, Fan Duty, Pump Duty, CDU outlet temperature, CDU inlet temperature, ambient temperature and total GPU power. Principal component analysis was then applied to reduce the data to two dimensions for easier screening. During outlier detection, the Mahalanobis distance of each sample was first calculated, and a threshold was set based on these distances. Samples exceeding this threshold were considered outliers and subsequently removed.

2.5 PSO method

The PSO algorithm is employed to optimize Fan Duty and Pump Duty, aiming to minimize the cooling system's energy consumption while maintaining the server's safe operating temperature.

Since the server cooling system involves multiple variables with nonlinear interactions, traditional control methods struggle to account for these interdependencies. PSO can simulate collective behaviors, iteratively exploring a wide solution space to identify the global optimum, effectively avoiding local minima, achieving precise energy efficiency optimization and enhancing system stability and adaptability. In this study, PSO is selected as the method for control parameter optimization, as it can explore the nonlinear relationship between Fan Duty and Pump Duty and find the optimal solution within a broad solution space. The complete algorithm flow is shown in Algorithm 2.1.



```
Algorithm 2.1: Particle Swarm Optimization (PSO)
  Input: Maximum iterations $iter\_max$, swarm size, and parameters
         $c_1$, $c_2$, and inertia weight $w$.
  Output: Global best position $gBest$.
Initialize particle swarm $P$ with random positions and velocities;
Set iteration counter $i = 1$;
while $i \leq iter\_max$ do
    for particle $p \in P$ do
        Evaluate $f(p)$;
        if $f(p)$ is better than $f(pBest)$ then
            Update $pBest = p$;
        end
    end
    Update $gBest$ as the best particle in $P$;
    foreach particle $p \in P$ do
        Update velocity:
        $v = w \cdot v + c_1 \cdot \mathrm{rand}() \cdot (pBest - p) + c_2 \cdot \mathrm{rand}() \cdot (gBest - p)$;
        Update position: $p = p + v$;
    end
    Increment $i = i + 1$;
end
return $gBest$
```

In the PSO algorithm, the inertia weight w, the cognitive learning factor $c1$, and the social learning factor $c2$ play a decisive role in the iteration performance. 2016 Wang [18] proposed using the (Number of Global Optimum Transcendences, Pnum) metric to quantify the swarm's exploration capability and identify the optimal combination of w, $c1$ and $c2$. This metric measures how many particles in each iteration generate solutions that surpass the previous generation's global best, thereby evaluating the swarm's overall search behavior and optimization effectiveness.

The mathematical definition of Pnum is as follows:

$$Pnum = \sum_{t=1}^{T} \sum_{I=1}^{N} I\left\{ f_i(t) < f_{gBest}(t-1) \right\}, \tag{2.3}$$

where $Pnum$ is the total number of times the entire particle swarm surpasses the previous best solution during the total iteration process, T is the total number of iterations, N is the number of particles, $f_i(t)$ represents the objective function value of the ith particle in the tth iteration, $f_{gBest}(t-1)$ is the global optimal target value of the previous iteration and $I(\cdot)$ is a Boolean function that takes on the value 1 if the condition is true and 0 otherwise.

The optimal hyperparameter combination for PSO should be determined based on the characteristics of the specific application system, as there is no universal solution that applies to all problems. When the objective function contains multiple local minima, it is generally preferable to increase the number of particles to enhance exploration diversity. In high-dimensional problems, the number of iterations should be extended to ensure sufficient global search depth. In this study, the control solution space of the server cooling system only involves the two dimensions of Fan Duty and Pump Duty, making it a low-dimensional search problem. However, under the temperature constraint, there exist multiple combinations of Fan Duty and Pump Duty that satisfy the constraints, causing the energy consumption term in the objective function to exhibit multimodality with numerous local minima. Based on these characteristics, this study adopts a "low iteration, high particle density" design principle to improve particle distribution coverage and accelerate the convergence efficiency of control decisions. The final configuration sets the maximum number of iterations to 10 and the number of particles to 30.

The solution space of application system in this study varies across different situation, so the benefit and feasibility of dynamic tuning the hyperparameters are uncertain and may complicate the problem. Thus, this study ultimately selected $w = 0.2$, $c1 = 1.4$ and $c2 = 1.4$ as the PSO hyperparameter combination. Under simulation conditions with a maximum number of 10 iterations, this combination achieved an average of $Pnum = 14$ optimal solutions, demonstrating excellent global search capabilities and stable evolutionary performance.

2.6 Traditional controller design-control group

To evaluate the energy-saving performance of the proposed LP Controller compared with existing control strategies, this study used the static constant-flow control mode described by Jifang (2021) [10] as the reference design for the comparison controller, coupled with a proportional-integral-derivative (PID) controller for water temperature regulation.

The comparison system adopts a constant-flow configuration, with the Pump Duty fixed at 70%. This value was derived based on the system design conditions: the pump's operational upper limit was set at 90%, while the maximum heating capacity under experimental conditions corresponded to 75%, giving a product of $0.9 \times 0.75 = 0.675$, which was rounded to 70% to meet the cooling requirements of the high-load constant-flow server system shown in Table 1. For water temperature control, the CDU outlet temperature was set to 40°C, following the internal water-cooling system standards of Kwang-Yun Machinery. The PID controller adjusts the Fan Duty to maintain the CDU outlet temperature close to this target, ensuring that the GPU1 Heater1 temperature remains below 70°C, which is defined as the safe operating condition in this study. During the experiments, the controller's response and steady-state outputs were observed, and the corresponding CDU energy consumption for Fan Duty and Pump Duty was recorded.

Based on the experimental results, two representative control combinations were selected for subsequent comparative analysis: the first corresponds to the peak control output, with Fan Duty and Pump Duty both at 70%, yielding a total CDU power consumption of 534.5 W; the second corresponds to the steady-state control output, with Fan Duty at 60% and Pump Duty at 70%, resulting in a total CDU power consumption of 475.4 W.

3. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

This section will analyze the practice of the control method in Section 2, including the analysis of the core model inside the controller.

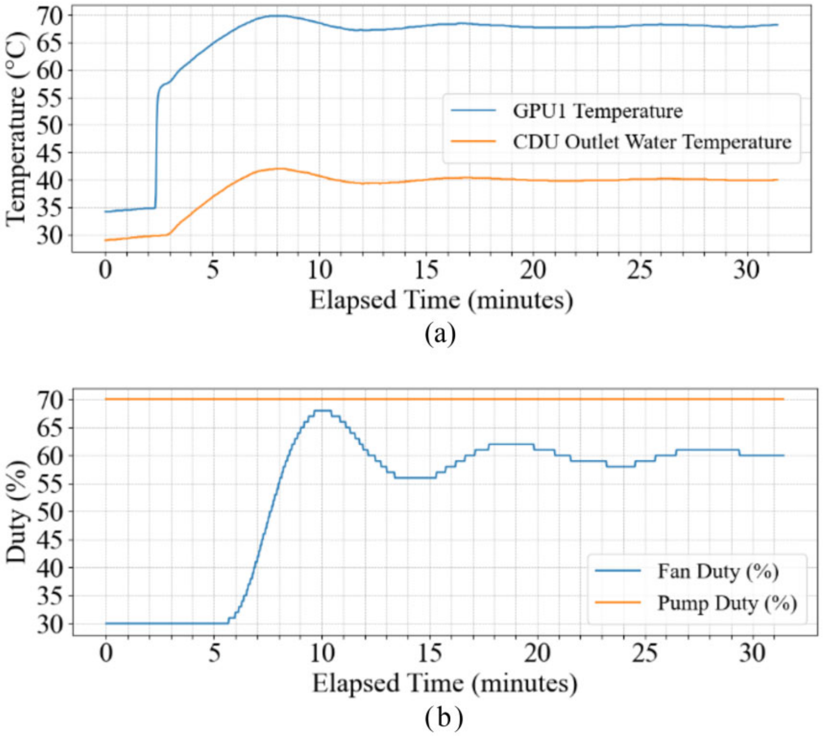


Figure 6 Response results of the control group. (a) GPU1 temperature response and CDU cooling trend. (b) Fan and Pump Duty cycle over time.

3.1 Traditional controller analysis

To ensure that the proposed LP Controller and the traditional controller operate under identical initial conditions during experimental comparisons, the experimental design was configured so that both controllers start from the same system state. Prior to controller activation, the GPU was set to a low-load mode, and the CDU outlet temperature was gradually raised to a predetermined initial value. Since the laboratory air-conditioning system caused the ambient temperature to fluctuate between 22 and 26°C, a continuous low-power heating of the cooling fluid by the GPU was applied to stabilize the CDU outlet temperature at 30°C, preventing environmental variations from affecting the initial condition. Once the system simultaneously satisfied both conditions—“GPU in the preset state (low load)” and “CDU outlet temperature stabilized at 30°C”—the traditional controller was activated, and the GPU was switched to high-load mode to simulate the controller’s response to a sudden increase in thermal load. Subsequent discussions of the LP Controller’s performance are also based on the same initial conditions and activation timing.

The response results and energy consumption analysis of the traditional controller are shown in Fig. 6.

After the reference controller was activated, as the GPU entered a high-load state, the temperature of GPU1 Heater1 rose rapidly, reaching a local peak between the 6th and 11th min, with a maximum temperature approaching 70°C. This temperature range did not exceed the safety limit defined in this study, indicating that the traditional controller possesses basic thermal stability control capability. The CDU outlet temperature also increased from the initial 30°C to ∼42°C, subsequently stabilizing within the range of 39.5–40.5°C. Regarding the control outputs, the Pump Duty was fixed at 70% throughout, simulating a constant-flow control logic, while the Fan Duty was adjusted via the PID controller.

In this experiment, the PID parameters were set as proportional gain KP = 5, integral gain KI = 0.02 and derivative gain KD = 0.01, remaining constant throughout the experiment. The initial Fan Duty was ∼30%, gradually increasing after activation and reaching a maximum of 68% around the 11th min. It then decreased and eventually stabilized at 60% after the 24th min. Based on the corresponding energy consumption curves of the control outputs, two representative control conditions were identified: the peak stage with Fan Duty = 68% and Pump Duty = 70%, corresponding to a total CDU power consumption of 534.5 W, and the steady-state stage with Fan Duty = 60% and Pump Duty = 70%, corresponding to a total CDU power consumption of 475.4 W. These two sets of control outputs and their associated energy consumption are used in this study as the benchmark for evaluating the performance of the LP Controller, providing a quantitative reference for energy-saving potential and dynamic stability.

3.2 LSTM model analysis

Figure 7 shows the scattered data points of fan duty and pump duty collected from actual machines.

Some data points can be observed outside the preset range. For instance, when Fan Duty is 80% and Pump Duty is 40%, this usually occurs under high GPU temperature conditions, triggering the system’s emergency cooling mechanism. In such cases, Fan Duty is automatically increased to 80% to enhance cooling,

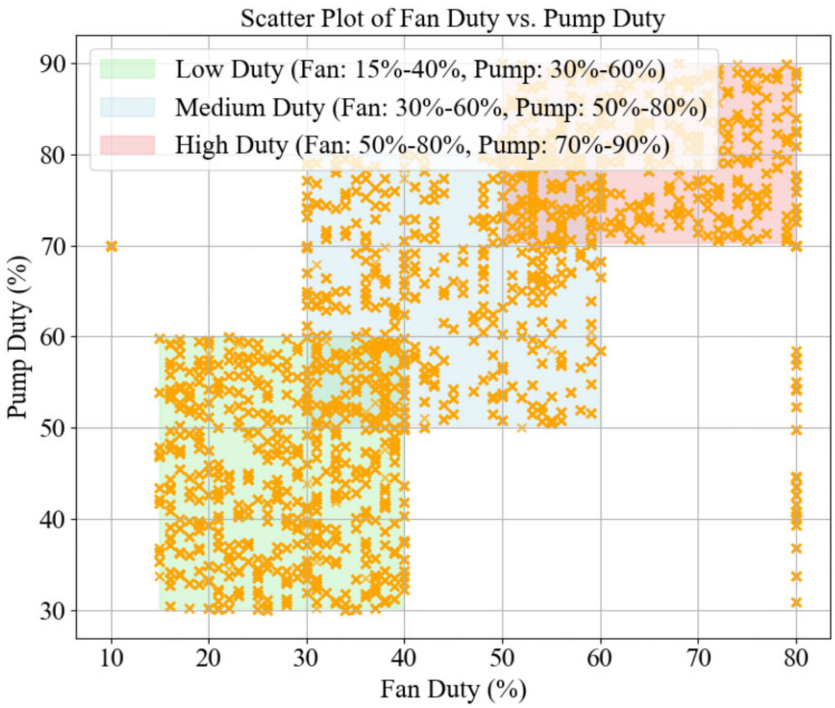


Figure 7 Training data collection Fan Duty and Pump Duty scatter plot.

while Pump Duty remains at its original setting, causing certain data points to exceed the predefined range.

The final results of the training and validation models are shown in Fig. 8. In Fig. 8, both consist of two subplots: (a) shows the temperature variation of GPU1 Heater1 over time, with the blue line representing the measured values and the orange line representing the LSTM model predictions; (b) is a scatter plot of measured vs. predicted values, with the red dashed line indicating the ideal diagonal, where points closer to the line indicate higher prediction accuracy.

For the validation set in Fig. 8, the errors are slightly higher: MAE = 0.5061, RMSE = 0.7538, max error = 6.8796, min error = 0.0019 and R^2 = 0.9278, but the overall predictions remain accurate and sufficient to support real-world temperature prediction tasks.

3.3 PSO hyperparameter design analysis

This study investigates the configuration of three key hyperparameters in the PSO algorithm, namely the inertia weight ww and the learning factors $c1$and $c2$. To ensure both algorithmic stability and practicality, a symmetric setting $c1 = c2$ is adopted, and the optimal parameter combinations are determined through theoretical analysis and large-scale simulations to enhance overall search efficiency and controller performance. A higher Pnum value indicates that the particle swarm more frequently generates solutions superior to the historical best during the exploration process, reflecting stronger exploration capability and evolutionary potential. The main advantage of using Pnum as a quantitative optimization metric is that it does not rely on the true global optimum, making it widely applicable to real-world control systems lacking explicit performance benchmarks. The overall experimental setup is summarized in Table 3.

Table 3 Experimental conditions for PSO hyperparameter settings.

Item	Description
Number of particles	30
Number of iterations	10
Repetition setting	Each group is executed 10 times and averaged
w	0.1–1.0; interval 0.1 → total 10 levels
$c1 = c2$	0.1–4.0; interval 0.1 → total 40 levels

In this study, a total of 400 hyperparameter combinations were designed: ww ranging from 0.1 to 1.0 with increments of 0.1 (10 values), and $c1 = c2$ ranging from 0.1 to 4.0 with increments of 0.1 (40 values). Each combination was repeated 10 times, resulting in a total of 4000 simulations shown in Fig. 9.

The average distribution of Pnum is shown in Fig. 10, presenting a three-dimensional surface plot for different combinations of the hyperparameters w and $c1 = c2$.

The experimental results indicate that the maximum average Pnum is 15.07, corresponding to the optimal parameter combination of $w = 0.2$ and $c1 = c2 = 1.4$. This combination represents the setting that most frequently generates particles superior to the historical best throughout the overall search process, demonstrating excellent exploration capability and stability. Therefore, this combination was adopted for subsequent control experiments.

To further understand the impact of individual parameters on algorithm performance, a univariate analysis was conducted for both w and $c1 = c2$. Figure 11 shows the average Pnum for all $c1 = c2$ combinations under different w values. Overall, Pnum exhibits a decreasing trend, with a relative peak around $w \approx 0.2$w, indicating that a lower inertia weight helps particles adjust

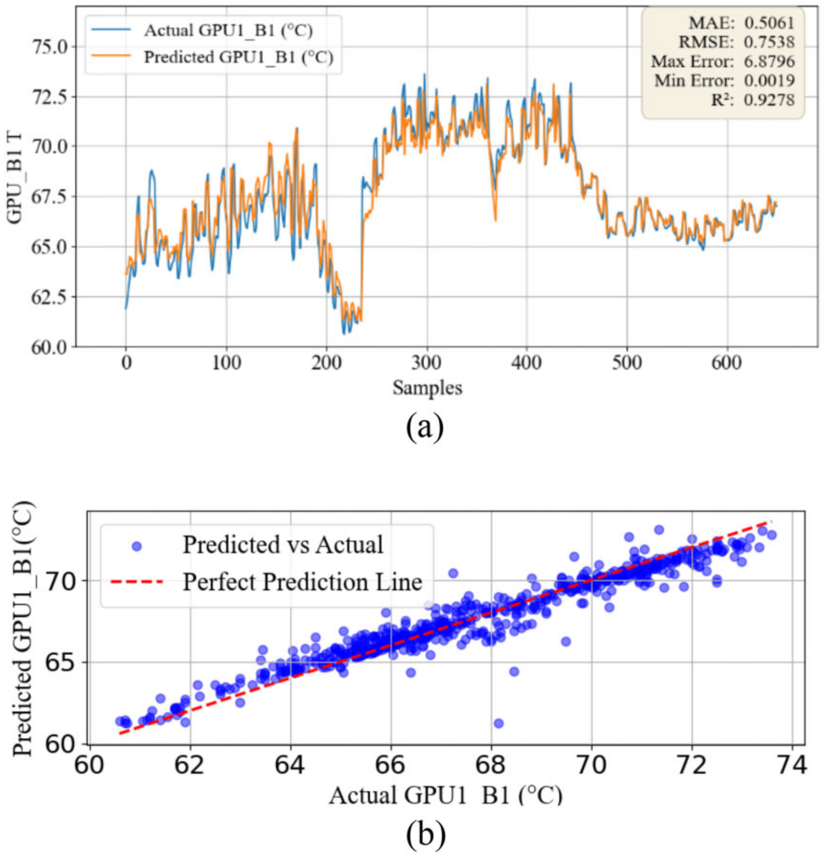


Figure 8 Validation data results graph. (a) Time series prediction: actual vs. predicted GPU1 B1 (validation set). (b) Real vs. predicted (validation set).

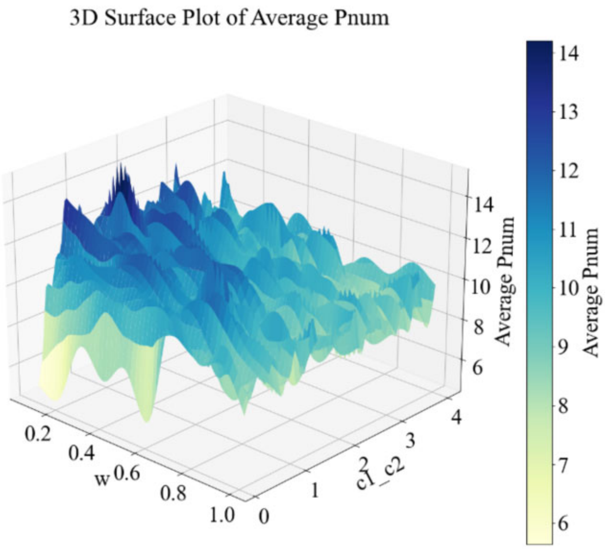


Figure 9 Distribution of average Pnum for different combinations of w and $c1 = c2$ (3D surface plot).

direction more quickly and increases the likelihood of escaping local optima.

Figure 11 presents the average Pnum under variations of $c1 = c2$, showing a sawtooth-like nonlinear distribution. In the range of $c1 = c2$ between 1.2 and 1.6, Pnum reaches a significant peak, reflecting that learning factors set too high or too low can suppress particle exploration flexibility, while a moderate value promotes the integration of individual and swarm best experiences. Combining the insights from Figs 10 and 11, it can be concluded that an excessively large inertia weight *www* causes particles to be overly conservative and difficult to deviate from existing directions, while $c1 = c2$ should be set within a mid-range to balance stability and activity. This also validates the rationality and representativeness of the optimal combination $w = 0.2$ and $c1 = c2 = 1.4$.

3.4 LP controller response result analysis

Figure 12 shows the control results of the LP Controller without introducing the penalty term into the objective function.

The controller is able to steadily maintain the GPU1 Heater1 temperature below 70°C, demonstrating high reliability in overall performance. However, observing its time-series response behavior, it can be seen that to simultaneously achieve both energy-saving and temperature stability goals, the controller frequently alternates between two control modes: (1) increasing Pump Duty while decreasing Fan Duty, and (2) increasing Fan Duty while decreasing Pump Duty. From the algorithmic theoretical perspective, both of these adjustment methods are reasonable, as increasing either control parameter effectively enhances GPU cooling, while decreasing either output can reduce CDU energy consumption. However, since the

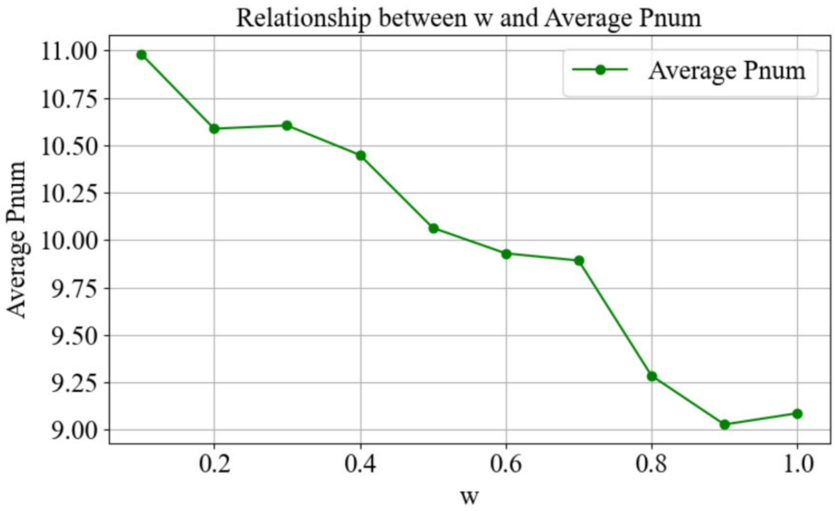


Figure 10 The effect of different w on the average Pnum (ignoring the effect of $c1 = c2$).

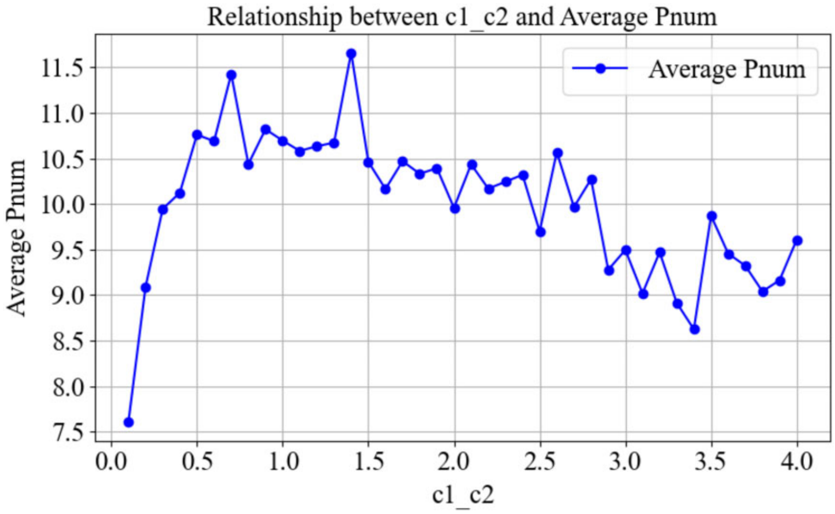


Figure 11 The effect of different $c1 = c2$ on the average Pnum (ignoring the w effect).

adjustment directions of the two parameters are opposite, this leads to an inconsistency in the modulation between Fan Duty and Pump Duty, reducing the coordination of control signals and further increasing the instantaneous fluctuation of the GPU temperature. This results in a slight delay in the cooling response and further amplifies the volatility of the Fan Duty and Pump Duty control outputs. To address this issue, this study introduces an additional "penalty term" in the objective function. The penalty term aims to reduce the occurrence of inconsistent directional changes between Fan Duty and Pump Duty at each control step, thereby improving the coordination of control outputs and enhancing the overall response stability. The response results after introducing the penalty term are shown in Fig. 13.

Figure 13 shows the continuous dynamic results of the LP Controller under high-load conditions over a period of 1.5 h.

During the initial control and steady-state phases, Fan Duty and Pump Duty generally exhibit a synchronized increase and decrease trend, demonstrating good coordinated control behavior and successfully avoiding the previously frequent alternating adjustments. The GPU temperature also became significantly more stable, remaining consistently at 70°C throughout the process with minimal fluctuations, and the CDU outlet temperature did not show any abnormal rise. During the high-load steady-state period, the average Fan Duty was 52.7% and the average Pump Duty was 51.2%.

3.5 Analysis of energy-saving results of high-load LP controller

To evaluate the energy-saving potential of the proposed LP Controller under high-load conditions, this section presents the measured CDU total power consumption during steady-state operation and compares it with the reference groups. The LP Controller, combining LSTM temperature prediction and PSO control parameter optimization, achieved an average Fan Duty of 52.7% and Pump Duty of 51.2% during the high-load steady-state period, resulting in a total CDU power consumption of only 320.7 W. This value represents the average

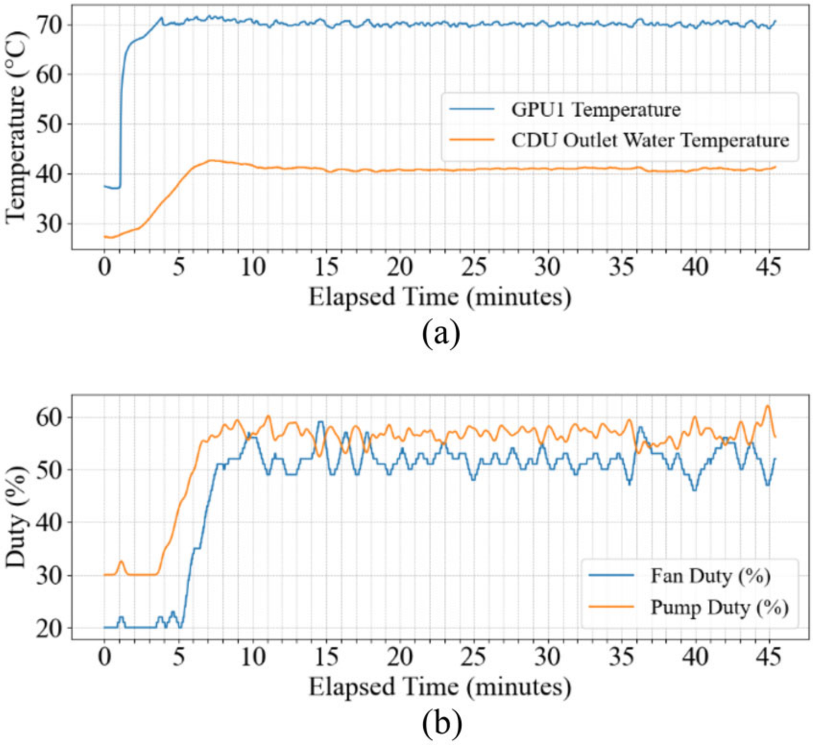


Figure 12 LP Controller does not introduce penalty response diagram. (a) GPU1 temperature response and CDU cooling trend. (b) Fan and Pump Duty cycle over time.

measured after running for a period under stable temperature control conditions, during which the GPU was under the high load set in this study. However, due to the small dynamic variations in actual GPU power consumption, this control output also exhibited normal fluctuations, so the data analysis was based on the average, reflecting actual system operating behavior. To verify the energy-saving effect, the results were compared with those of two traditional control strategies: the first group, during the peak control output stage, where Fan Duty = 70% and Pump Duty = 70%, resulting in a total CDU power consumption of 534.5 W; and the second group, during the steady-state control output stage, where Fan Duty = 60% and Pump Duty = 70%, resulting in a total CDU power consumption of 475.4 W. The comparison results show that the LP Controller, under high-power load, reduced the CDU total power consumption by 213.8 W compared to reference group one, achieving an energy-saving rate of 39.9%, and by 154.7 W compared to reference group two, achieving an energy-saving rate of 32.5%. The energy-saving effect is significant, as shown in Fig. 14.

This study focuses on energy-saving analysis under maximum thermal load conditions for two main reasons: (1) high-load scenarios are the most challenging and energy-intensive situations in data centers, thus offering the greatest energy-saving potential, and (2) in medium- and low-load conditions, the traditional controller uses PID to control Fan Duty while keeping Pump Duty fixed, meaning the energy-saving benchmarks for CDU outlet temperature and flow rate need to be re-established, and corresponding comparison conditions must be redesigned, which exceeds the scope of this study.

4. CONCLUSION

This study focuses on minimizing the cooling system energy consumption of liquid-cooled servers under high thermal load conditions while ensuring the safe operation of the GPU temperature, thereby improving the energy efficiency of the cooling system. To achieve this goal, this study combines LSTM and PSO to propose a controller (LP Controller) based on deep learning and swarm intelligence algorithms, which is successfully applied to the physical CDU system control architecture. The LSTM is responsible for GPU temperature prediction, capturing the temporal delays and nonlinear relationships between the cooling system and temperature changes. PSO dynamically optimizes Fan Duty and Pump Duty, utilizing a multi-objective function design that balances GPU thermal safety and cooling energy efficiency. This study has resulted in several important conclusions:

1. The prediction model trained with LSTM successfully captures the impact of factors such as fans, pumps and ambient temperature on GPU temperature. It can predict the GPU temperature at the next sampling moment, providing a basis for the controller to adjust control parameters in advance, improving the system's dynamic response speed and control stability while also achieving energy savings.
2. Design a dual-objective function and introduce a penalty term to suppress the inconsistent change direction of the fan and pump, thereby improving the stability of the control output.

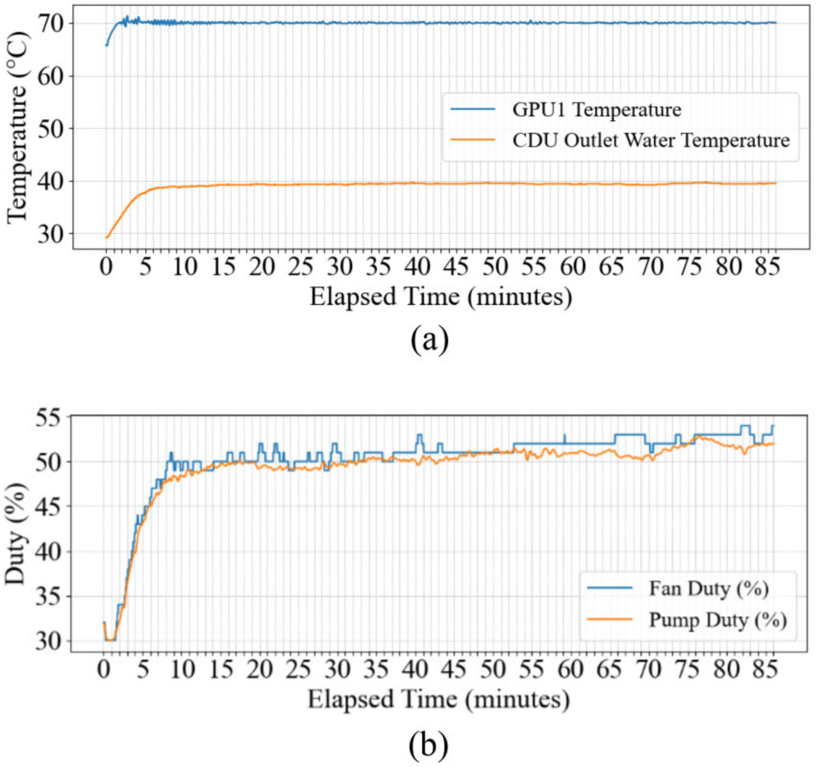


Figure 13 LP Controller. (a) GPU1 temperature response and CDU cooling trend. (b) Fan and Pump Duty cycle over time.

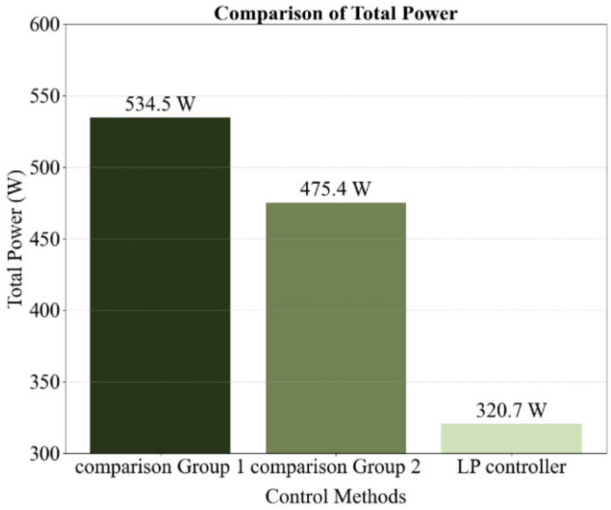


Figure 14 Comparison of total CDU power consumption between the LP Controller and the control group.

3. Use Pnum to select the best hyperparameter combination ($w = 0.2, c1 = c2 = 1.4$) to make the control output easier to escape from the local optimal solution and the output value is more reliable.
4. In terms of energy efficiency, compared to the traditional control group, which controlled output peaks at 70% fan duty and 70% pump duty, the CDU system's total power consumption dropped from a maximum of 534.5 to 320.7 W, achieving a maximum energy saving of 39.9% using the LP Controller. During the control process, average fan duty and pump duty values were 52.7 and 51.2%, respectively, demonstrating stable and energy-efficient performance.

ACKNOWLEDGMENT

The authors would like to thank the partially support by Inventec Cooperation, TAIWAN, project, named ⌊Research of Control System for Liquid-Cooled Server Cooling Systems⌋.

AUTHOR CONTRIBUTIONS

Hsuan-Li Shih (Formal Analysis, Investigation, Methodology), Rongshun Chen (Investigation, Supervision, Writing – review & editing)

REFERENCES

1. Qaisar F, Shahab H, Iqbal M, Sargana EH, Aqeel E, Qayyum M. Recent trends in cloud computing and IoT platforms for its management and development: a review. *Pakistan Journal of Engineering and Technology* 2023;**6**(1):98–105.
2. G. D. C. Authority. Uptime institute global data center survey 2024. 2024. [Online]. Available: https://datacenter.uptimeinstitute.com/rs/711RIA145/images/2024.GlobalDataCenterSurvey.Report.pdf?version=0, accessed 2024-11-21.
3. Pilz K, Heim L. Compute at scale: a broad investigation into the data center industry. *arXiv* preprint arXiv:2311.02651.
4. Koot M, Wijnhoven F. Usage impact on data center electricity needs: a system dynamic forecasting model. *Applied Energy* 2021;**291**:116798.

5. Ahmed KMU, Bollen MHJ, Alvarez M. A review of data centers energy consumption and reliability modeling. *IEEE Access* 2021;**9**: 152 536–152 563.
6. He W, Ding S, Zhang J, Pei C, Zhang Z, Wang Y, Li H. Performance optimization of server water cooling system based on minimum energy consumption analysis. *Applied Energy* 2021;**303**:117620.
7. Athavale J, Yoda M, Joshi Y. Comparison of data driven modeling approaches for temperature prediction in data centers. *International Journal of Heat and Mass Transfer* 2019;**135**:1039–1052.
8. Asgari S, MirhoseiniNejad S, Moazamigoodarzi H, Gupta R, Zheng R, Puri IK. A gray-box model for real-time transient temperature predictions in data centers. *Applied Thermal Engineering* 2021;**185**:116319.
9. Ham S-W, Kim M-H, Choi B-N, Jeong J-W. Simplified server model to simulate data center cooling energy consumption. *Energy and Buildings* 2015;**86**:328–339.
10. Moazamigoodarzi H, Gupta R, Pal S, Tsai PJ, Ghosh S, Puri IK. Modeling temperature distribution and power consumption in it server enclosures with row-based cooling architectures. *Applied Energy* 2020;**261**:114355.
11. Asgari S, Moazamigoodarzi H, Tsai PJ, Pal S, Zheng R, Badawy G, Puri IK. Hybrid surrogate model for online temperature and pressure predictions in data centers. *Future Generation Computer Systems* 2021;**114**:531–547.
12. Zapater M, Risco-Martín JL, Arroba P, Ayala JL, Moya JM, Hermida R. Runtime data center temperature prediction using grammatical evolution techniques. *Applied Soft Computing* 2016;**49**:94–107.
13. Verma P, Reddy SV, Ragha L, Datta D. Comparison of time-series forecasting models. *2021 International Conference on Intelligent Technologies (CONIT)*, 2021, 1–7.
14. Sonata I, Heryadi Y. Comparison of LSTM and transformer for time series data forecasting. *2024 7th International Conference on Informatics and Computational Sciences (ICICoS)*, 2024, 491–495.
15. 戴仲瑜. 灰狼演算法應用於複合電力電動車輛系統之控制器設計. Master's thesis, National Taiwan Normal University (Taiwan), 2020.
16. Jiang M-Q. 鯨魚演算法應用於三動力複合動力系統之最佳化能量管理. Master's thesis, National Taiwan Normal University (Taiwan), 2021.
17. Gad AG. Particle swarm optimization algorithm and its applications: a systematic review. *Archives of Computational Methods in Engineering* 2022;**29**(5):2531–2561.
18. 王东风, 孟丽. 粒子群优化算法的性能分析和参数选择. *自动化学报* 2016;**42**(10):1552–1561.

Received: 30 September 2025; **Revised:** 15 November 2025; **Accepted:** 20 November 2025